

AMENDMENTS TO THE CLAIMS

Upon entry of this amendment, the following listing of claims will replace all prior versions and listings of claims in the pending application.

IN THE CLAIMS

Please amend claims 17 and 32 and add claim 44 as follows:

1. (Previously Presented) A computer-implemented method for adaptively filtering messages routed across a network by generating exception rules to rejection rules based on attributes of messages previously received and rejected, the method comprising:

receiving a first message;

rejecting the first message based on a rejection rule;

determining, for the first message, an attribute that triggered the rejection rule;

incrementing, for the attribute, a count of the number of messages rejected based on the attribute;

based on the count for the attribute, determining a frequency with which messages having the attribute were rejected based on the rejection rule;

generating an exception rule to the rejection rule which rejected the messages with the attribute, responsive to the determined frequency exceeding a threshold;

receiving a second message having the attribute; and

allowing the second message, responsive to the exception rule.

2. (Cancelled).

3. (Cancelled).

4. (Original) The method of claim 1, wherein the attribute is one of a message component, a value, a data type, and a length.

5. (Original) The method of claim 1, wherein the frequency is a weighted count of occurrences of the attribute.

6. (Original) The method of claim 1, wherein the frequency is a direct count of occurrences of the attribute.

7. (Original) The method of claim 1, wherein the rejected messages are URL requests, each URL request having at least one URL component, the method further comprises: maintaining a frequency for each instance of a URL component, wherein the frequency is a function of a number of occurrences with which a URL component and its descendants were rejected by a rule; selecting a URL component according to a set of constraints; and generating an exception rule for the selected URL component and its descendants.

8. (Original) The method of claim 7, wherein the exception rule is generated by inferencing a scalar data type of the descendants of the selected URL component.

9. (Original) The method of claim 7, wherein the set of constraints is selecting a URL component with a frequency exceeding a threshold and having no children with a frequency above the threshold.

10. (Original) The method of claim 7, wherein the set of constraints is selecting a URL component with the frequency exceeding a threshold.

11. (Original) The method of claim 7, wherein the function is an aggregate of a number of occurrences with which the URL component was rejected by a rule and the number of occurrences with which descendants of the URL component were rejected by the rule.

12. (Original) The method of claim 1, wherein the messages are URL requests, each URL request having at least one URL component, and the method further comprises: storing rejected URLs in a trie structure, wherein each node in the trie structure is associated with a URL component; maintaining a frequency for each node associated with a URL component, wherein the frequency is a function of a number of occurrences with which a URL component associated with a node and its descendants were rejected with a rule; selecting a node in the trie structure

according to a set of constraints; and generating an exception rule for the selected node and its descendants.

13. (Original) The method of claim 12, wherein the exception rule is generated by inferencing a scalar data type of the descendants of the selected node.

14. (Original) The method of claim 12, wherein the set of constraints is selecting a node with the frequency exceeding a threshold.

15. (Original) The method of claim 12, wherein the set of constraints is selecting a node with a frequency exceeding a threshold and having no children with a frequency above the threshold.

16. (Original) The method of claim 1, wherein the threshold is a product of a total number of messages over a time interval and a percentage of the messages that should be allowed to pass.

17. (Currently Amended) A system for adaptively filtering messages routed across a network by generating exception rules to rejection rules based on attributes of messages previously received and rejected, the system comprising:

- a receiver which receives a first message;

- a filter which rejects the first message based on a rejection rule;

- a learning engine, for extracting, from the first message, an attribute of the message that triggered the rejection rule, for incrementing, for the attribute, a count of the number of messages rejected based on the attribute, for determining, based on the count for the attribute, a frequency with which messages having the attribute were rejected based on the rejection rule, and for generating an exception rule to the rejection rule which rejected the messages with the attribute, responsive to the determined frequency exceeding a threshold; and ~~wherein~~ wherein

- the filter applies the exception rule to subsequent messages to determine whether to allow the subsequent messages.

18. (Original) The system of claim 17, wherein the filter is further adapted to: identify an attribute of the rejected message that triggered a rejection rule; for the triggered rejection rule, identify an exception rule that matches the attribute; and . apply the exception rule to a rejected message to determine whether to allow the message.

19. (Original) The system of claim 17, wherein the attribute is one of a value, data type, and length.

20. (Original) The system of claim 17, wherein the frequency is a weighted count of occurrences of the attribute.

21. (Original) The system of claim 17, wherein the frequency is a direct count of occurrences of the attribute.

22. (Original) The system of claim 17, wherein the rejected messages are URL requests, each URL request having at least one URL component, and the learning engine further adapted to: maintain a frequency for each instance of a URL component, wherein the frequency is a function of a number of occurrences with which a URL component and its descendants were rejected with a rule; select a URL component according to a set of constraints; and generate an exception rule for the selected URL component and its descendants.

23. (Original) The system of claim 17, wherein the set of constraints is selecting a URL component with the frequency exceeding a threshold and having no children with a frequency above the threshold.

24. (Original) The system of claim 17, wherein the set of constraints is selecting a URL component with the frequency exceeding a threshold.

25. (Original) The system of claim 22, wherein the function is an aggregate of a number of occurrences with which the URL component was rejected by a rule and a number of occurrences with which descendants of the URL component were rejected by the rule

26. (Original) The system of claim 22, wherein the exception rule is generated by inferencing a scalar data type of the descendants of the selected URL component.

27. (Original) The system of claim 17, wherein the messages are URL requests, each URL request having at least one URL component, and the learning engine is further adapted to: store rejected URLs in a trie structure, wherein each node in the trie structure is associated with a URL component; maintain a frequency for each node associated with a URL component, wherein the frequency is a function of a number of occurrences with . which a URL component and its descendants were rejected with a rule; select a node according to a set of constraints; and generate an exception rule for the selected node and its descendants.

28. (Original) The system of claim 17, wherein the set of constraints is selecting a node with the frequency exceeding a threshold.

29. (Original) The system of claim 17, wherein the set of constraints is selecting a node with a frequency exceeding a threshold and having no children with a frequency above the threshold.

30. (Original) The system of claim 27, wherein the exception rule is generated by inferencing a scalar data type of the descendants of the selected node.

31. (Original) The system of claim 17, wherein the threshold is a product of a total number of messages over a time interval and a percentage of the messages that should be allowed to pass.

32. (Currently amended) A computer program product comprising: a computer-readable medium having computer program code embodied therein for adaptively filtering messages routed across a network by generating exception rules to rejection rules based on attributes of messages previously received and rejected, the computer program code adapted to:

receive a first message;

reject the first message based on a rejection rule;

determine, for the first message, an attribute that triggered the rejection rule;

~~extract, from the rejected message, an attribute that triggered the rejection rule;~~

increment, for the attribute, a count of the number of messages rejected based on the attribute;

based on the count for the attribute, determining a frequency with which messages having the attribute were rejected based on the rejection rule; and

generate an exception rule to the rejection rule which rejected the message with the attribute, responsive to the determined frequency ~~of the attribute~~ exceeding a threshold;

receiving a second message having the attribute; and

allowing the second message based on the exception rule.

33. (Cancelled)

34. (Cancelled).

35. (Original) The computer program product of claim 32, wherein the attribute is one of a value, data type, and length.

36. (Original) The computer program product of claim 32, wherein the frequency is a weighted count of the occurrences of the attribute.

37. (Original) The computer program product of claim 32, wherein the frequency is a direct count of the occurrences of the attribute.

38. (Original) The computer program product of claim 32, wherein the rejected messages are URL requests, each URL request having at least one URL component, wherein the computer program code is further adapted to: maintain a frequency for each instance of a URL component, wherein the frequency is a function of a number of occurrences with which a URL component and its descendants were rejected by a rule; select a URL component with the frequency exceeding a threshold and having no children with a frequency above the threshold; and generate an exception rule for the selected URL component and its descendants.

39. (Original) The computer program product of claim 32, wherein the computer program code is further adapted to generate the exception rule by inferencing a scalar data type of the descendants of the selected URL component.

40. (Original) The computer program product of claim 38, wherein the function is an aggregate of a number of occurrences with which the URL component was rejected by a rule and the number of occurrences with which descendants of the URL component were rejected by the rule.

41. (Original) The computer program product of claim 32, wherein the rejected messages are URL requests, each URL request having at least one URL component, wherein the computer program code is further adapted to: store rejected URLs in a trie structure, wherein each node in the trie structure is associated with a URL component; maintain a frequency for each node associated with a URL component, wherein the frequency is a function of a number of occurrences with which a URL component and its descendants were rejected with a rule; select a node with the frequency exceeding a threshold; and generate an exception rule for the selected node and its descendants.

42. (Original) The computer program product of claim 32, wherein the computer program code is further adapted to generate the exception rule by inferencing a scalar data type of the descendants of the selected node.

43. (Original) The computer program product of claim 32, wherein the function is an aggregate of a number of occurrences with which the URL component was rejected by a rule and a number of occurrences with which descendants of the URL component were rejected with the rule.

44. (Original) The computer program product of claim 32, wherein the computer program code is further adapted to determine the threshold as a product of a total number of messages over a time interval and a percentage of the messages that should be allowed to pass.

45. (New) The method of claim 1, wherein the step of receiving a second message having the attribute further comprises:

rejecting the second message based on the rejection rule; and
determining that the exception rule to the rejection rule exists.